

FIG. 1.

# FIG. 20 "OTHERS"

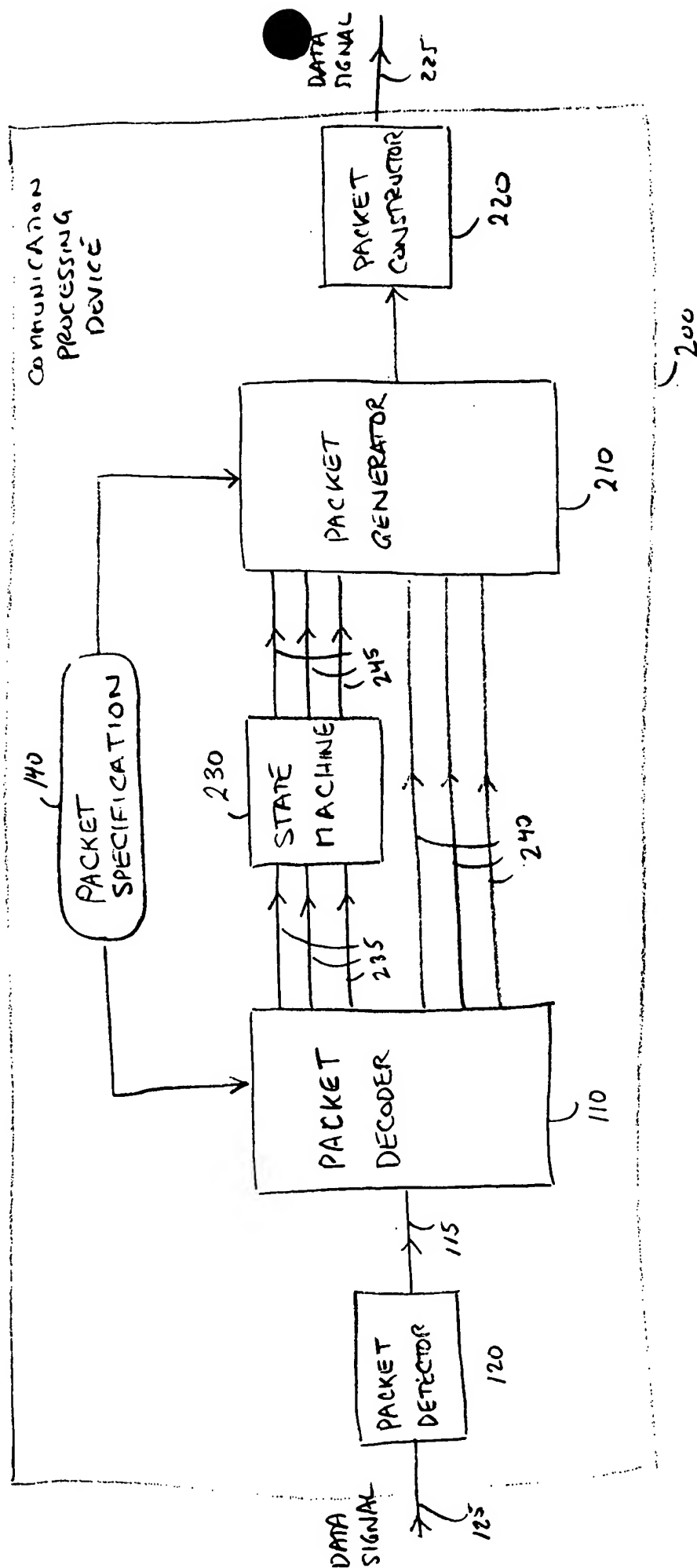


FIG. 2

202207040800T 10085440 02300

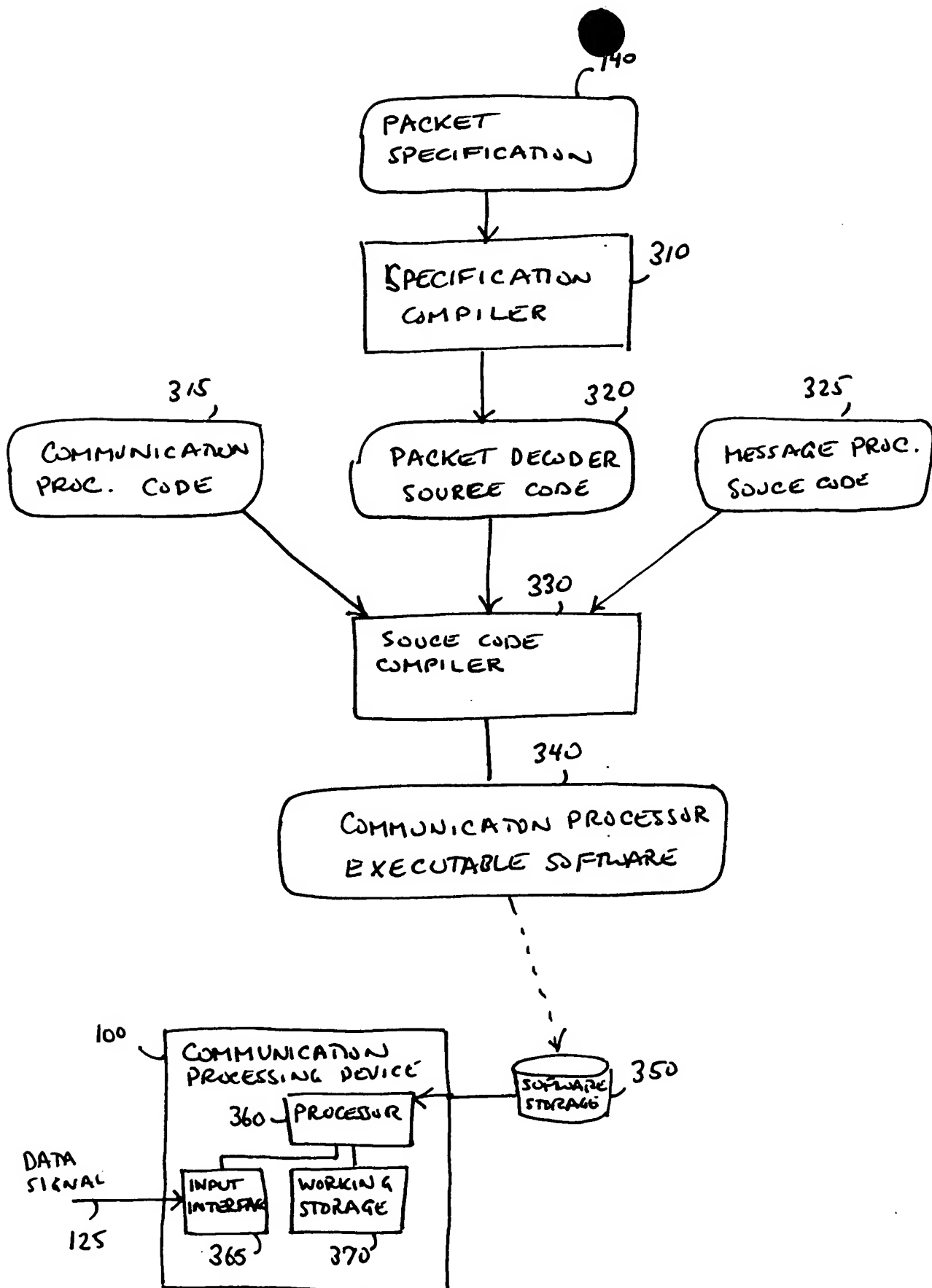


FIG. 3

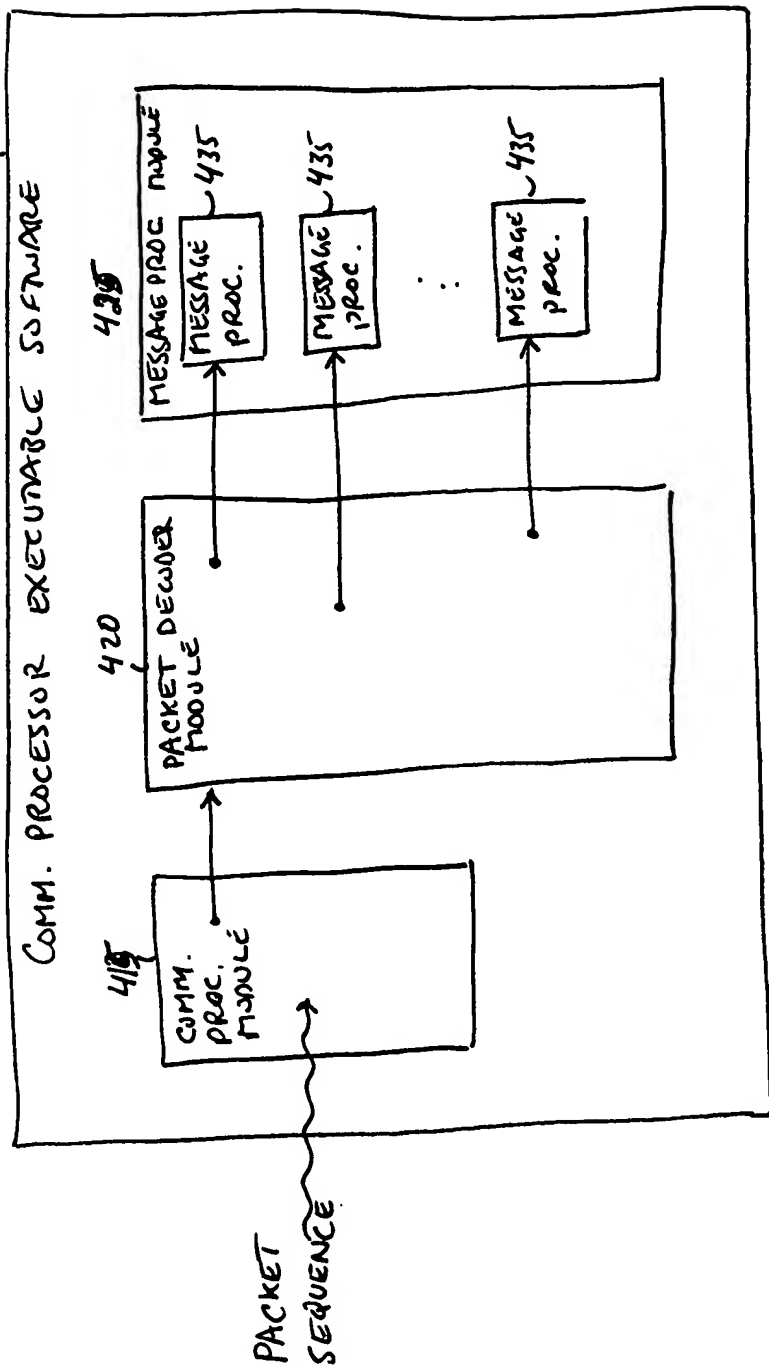


FIG. 4

10035440-023302

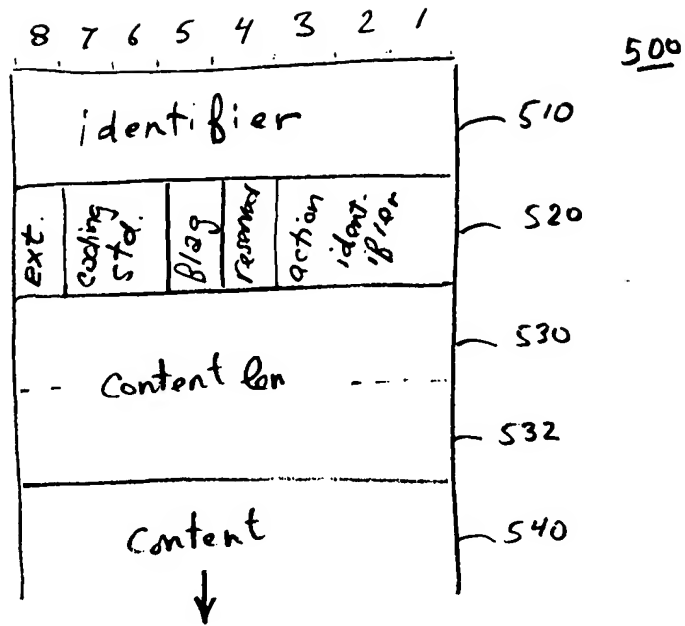


FIG. 5

```

100 packet ("information element") {
101     field(:"identifier",8) {alt{
102         0x04:"narrow-band bearer capability"
103         0x08:"cause" 0x14:"call state"
104         0x1e:"progress indicator"
105     }
106     .
107     .
108     .
109     0x5c:"quality of service parameter"
110     0x5e:"broad-band bearer capability"
111     0x5f:"broad-band low layer information"
112     .
113     .
114     .
115     0x7c:"narrow-band low layer compatibility"
116     0x7d:"narrow-band high layer compatibility"
117     rest:"unknown"
118 }
119
120 field(:"ext bit")
121 field(:"coding standard",2) {alt{
122     0:"ITU-T standardized coding"
123     1:"ISO/IEC standard"
124     2:"national standard"
125     3:"standard defined for the network present on the network instruction field is
126         not significant"
127 }
128
129 field(:"flag")
130 field(:"reserved") {"should be 0x00"}
131 field(:"action identifier",3)
132 len field("contens len",16)

```

FIG. 6A

## 20320" 045303"

```

144 alt(var("identifier")) {
145     0x04:packet("narrow-band bearer capability") // Q.2931 pg. 79
146 // Q.2931 pg. 64
147     0x08:packet("cause") // Q.2931 pg. 68
148     0x14:packet("call state") // Q.2931 pg. 59
149     0x1e:packet("progress indicator") // Q.2931 pg. 81
.
.
.
156     0x5c:packet("quality of service parameter") // Q.2931 pg. 72
157     0x5e:packet("broad-band bearer capability") // Q.2931 pg. 51
158     0x5f:packet("broad-band low layer information") // Q.2931 pg. 54
.
.
.
173     0x7c:packet("narrow-band low layer compatibility") // Q.2931 pg. 80
174     0x7d:packet("narrow-band high layer compatibility") // Q.2931 pg. 79
175     rest:"unknown"
176 }
177 }
```

FIG. 6B

40085440-022802

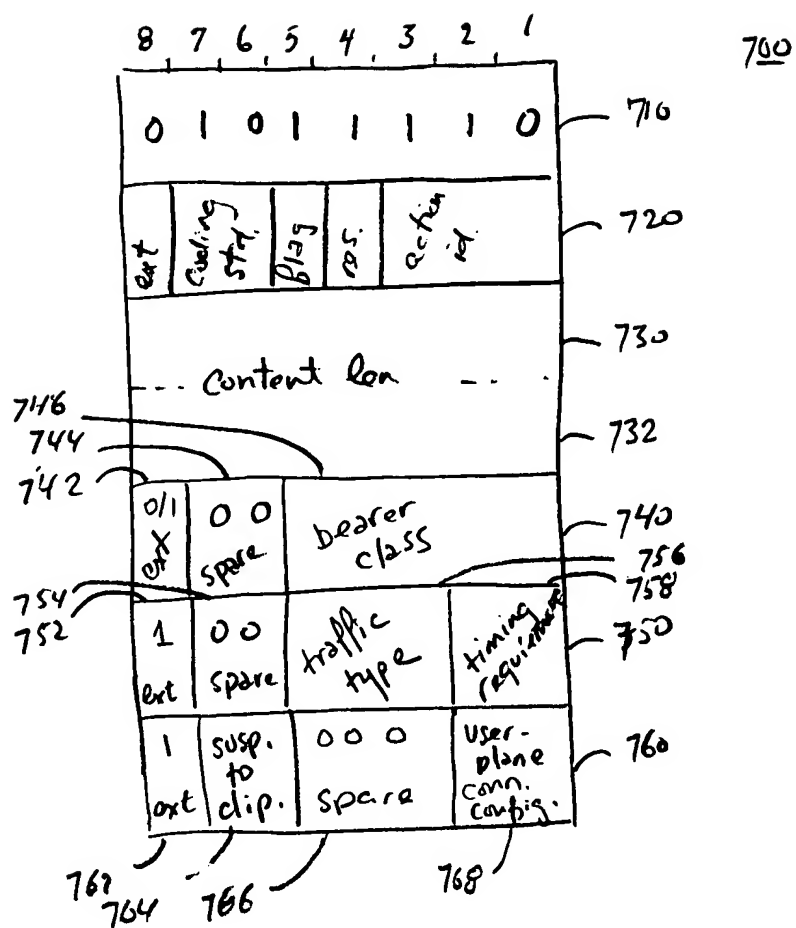


FIG. 7



```

100 packet("broad-band bearer capability") {
101   field(:"ext bit")
102   field(:"spare", 2)
103   field("bearer class",5) {alt{
104     1:"BCOB-A" 3:"BCOB-C" 0x10:"BCOB-X" rest:"reserved"}}}
105   alt(var(:"ext bit")) {
106     0: field(:"ext bit") field(:"spare",2)
107       field("traffic type",3) {alt{
108         0:"no indication"
109         1:"constant bit rate"
110         2:"variable bit rate"
111         rest:"reserved"}}}
112     field("timing requirements",2) {alt{
113       0:"no indication"
114       1:"end-to-end timing required"
115       2:"end to end timing not required"
116       rest:"reserved"}}}
117   }
118   field(:"ext bit")
119   field("susceptibility to clipping",2) {alt{
120     0:"not susceptible to clipping"
121     1:"susceptible to clipping"
122     rest:"reserved"}}}
123   field(:"spare",3)
124   field("user plane connection configuration",2) {alt{
125     0:"point-to-point"
126     1:"point-to-multipoint"
127     rest:"reserved"}}}
128 }

```

FIG. 8

```

100 void CQ_2931_Signaling:PP$information_element (char* aname, unsigned a) {
101     if( !CurPduLen() ) return;
102     ProcessPacketBeg( aname, a );
103
104     V$FP$identifier = ProcessField ( "identifier", 8);
105     FP$identifier ( V$FP$identifier );
106     V$FP$ext_bit = ProcessField ( "ext bit", 1);
107     FP$ext_bit ( V$FP$ext_bit );
108     V$FP$coding_standard = ProcessField ( "coding standard", 2);
109     FP$coding_standard ( V$FP$coding_standard );
110     V$FP$flag = ProcessField ( "flag", 1);
111     FP$flag ( V$FP$flag );
112     V$FP$reserved = ProcessField ( "reserved", 1);
113     FP$reserved ( V$FP$reserved );
114     V$FP$action_identifier = ProcessField ( "action identifier", 3);
115     FP$action_identifier ( V$FP$action_identifier );
116     V$FP$information_element$contents_len = ProcessField ( "contents len", 16);
117     FP$information_element$contents_len ( V$FP$information_element$contents_len );
118     SetPduLen ( V$FP$information_element$contents_len );
119
120     if ( V$FP$identifier == 0x04 ) {
121         PP$narrow_band_bearer_capability ( "narrow-band bearer capability", CurPduLen());
122     } else
123     if ( V$FP$identifier == 0x08 ) {
124         PP$cause ( "cause", CurPduLen());
125     } else
126     if ( V$FP$identifier == 0x14 ) {
127         PP$call_state ( "call state", CurPduLen());
128     }
129
130

```

FIG. 9A

```

131     } else
132     if ( V$FP$identifier == 0x1e ) {
133         PP$progress_indicator ( "progress indicator", CurPduLen() );
134
135     .
136     .
137     .
138
139     } else
140     if ( V$FP$identifier == 0x5c ) {
141         PP$quality_of_service_parameter ( "quality of service parameter", CurPduLen() );
142
143     } else
144     if ( V$FP$identifier == 0x5e ) {
145         PP$broad_band_bearer_capability ( "broad-band bearer capability", CurPduLen() );
146
147     } else
148     if ( V$FP$identifier == 0x5f ) {
149         PP$broad_band_low_layer_information ( "broad-band low layer information",
150         CurPduLen() );
151
152     .
153     .
154     .
155
156     } else
157     if ( V$FP$identifier == 0x7c ) {
158         PP$narrow_band_low_layer_compatibility ( "narrow-band low layer compatibility",
159         CurPduLen() );
160

```

FIG. 9B

203220 "0445804"

```
161 } else
162 if ( V$FP$identifier == 0x7d ) {
163     PP$narrow_band_high_layer_compatibility ( "narrow-band high layer compatibility",
164     CurPduLen());
165
166 } else {
167     PrintName("unknown");
168 };
169 ProcessPacketEnd(aname, a );
170 } // end of packet processor function definition PP$information_element
```

FIG. 9C

```

171 void CQ_2931_Signaling::FP$identifier (unsigned a) {
172
173     PrintName(" : ");
174
175     if ( a == 0x04 ) {
176         PrintName("narrow-band bearer capability");
177     } else
178     if ( a == 0x08 ) {
179         PrintName("cause");
180     } else
181     if ( a == 0x14 ) {
182         PrintName("call state");
183     } else
184     if ( a == 0x1e ) {
185         PrintName("progress indicator");
186
187         .
188         .
189         .
190
191     } else
192     if ( a == 0x5c ) {
193         PrintName("quality of service parameter");
194     } else
195     if ( a == 0x5e ) {
196         PrintName("broad-band bearer capability");
197     } else
198     if ( a == 0x5f ) {
199         PrintName("broad-band low layer information");
200

```

208220" 04458007

```
201 .
202 .
203 .
204
205     } else
206     if ( a == 0x7c ) {
207         PrintName("narrow-band low layer compatibility");
208     } else
209     if ( a == 0x7d ) {
210         PrintName("narrow-band high layer compatibility");
211     } else {
212         PrintName("unknown");
213     } ;
214 } // end of field processor function definition FP$identifier
```

FIG. 9E

```

215 void CQ_2931_Signaling::PP$broad_band_bearer_capability (char* aname, unsigned a) {
216     if( !CurPduLen() ) return;
217     ProcessPacketBeg( aname, a );
218
219     V$FP$ext_bit = ProcessField ( "ext bit", 1);
220     FP$ext_bit ( V$FP$ext_bit );
221     V$FP$spare = ProcessField ( "spare", 2);
222     FP$spare ( V$FP$spare );
223     V$FP$broad_band_bearer_capability$bearer_class = ProcessField ( "bearer class", 5);
224     FP$broad_band_bearer_capability$bearer_class (
225     V$FP$broad_band_bearer_capability$bearer_class );
226
227     if ( V$FP$ext_bit == 0 ) {
228         V$FP$ext_bit = ProcessField ( "ext bit", 1);
229         FP$ext_bit ( V$FP$ext_bit );
230         V$FP$spare = ProcessField ( "spare", 2);
231         FP$spare ( V$FP$spare );
232         V$FP$broad_band_bearer_capability$traffic_type = ProcessField ( "traffic type", 3);
233         FP$broad_band_bearer_capability$traffic_type (
234             V$FP$broad_band_bearer_capability$traffic_type );
235         V$FP$broad_band_bearer_capability$timing_requirements = ProcessField ( "timing
236             requirements", 2);
237         FP$broad_band_bearer_capability$timing_requirements (
238             V$FP$broad_band_bearer_capability$timing_requirements );
239
240     } else ;
241         V$FP$ext_bit = ProcessField ( "ext bit", 1);
242         FP$ext_bit ( V$FP$ext_bit );
243         V$FP$broad_band_bearer_capability$susceptibility_to_clipping = ProcessField (
244             "susceptibility to clipping", 2);

```

FIG. 9F

```

245 FP$broad_band_bearer_capability$susceptibility_to_clipping (
246 V$FP$broad_band_bearer_capability$susceptibility_to_clipping );
247 V$FP$spare = ProcessField ( "spare", 3 );
248 FP$spare ( V$FP$spare );
249 V$FP$broad_band_bearer_capability$user_plane_connection_configuration =
250 ProcessField ( "user plane connection configuration", 2 );
251 FP$broad_band_bearer_capability$user_plane_connection_configuration
252     (V$FP$broad_band_bearer_capability$user_plane_connection_configuration );
253
254 ProcessPacketEnd(aname, a );
255 }
256 // end of packet processor function definition PP$broad_band_bearer_capability

```

FIG. 9G



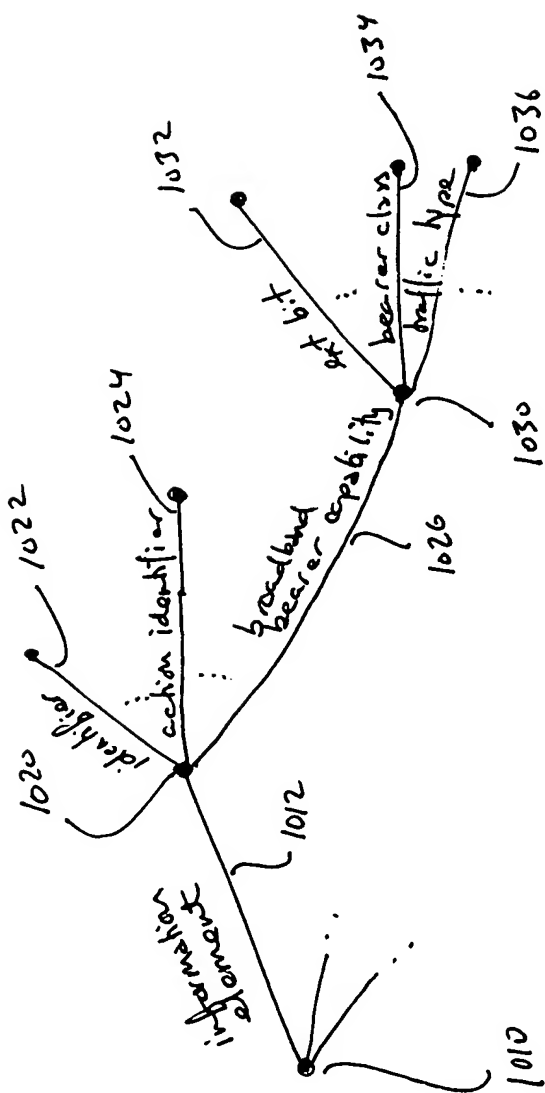


FIG. 10

10

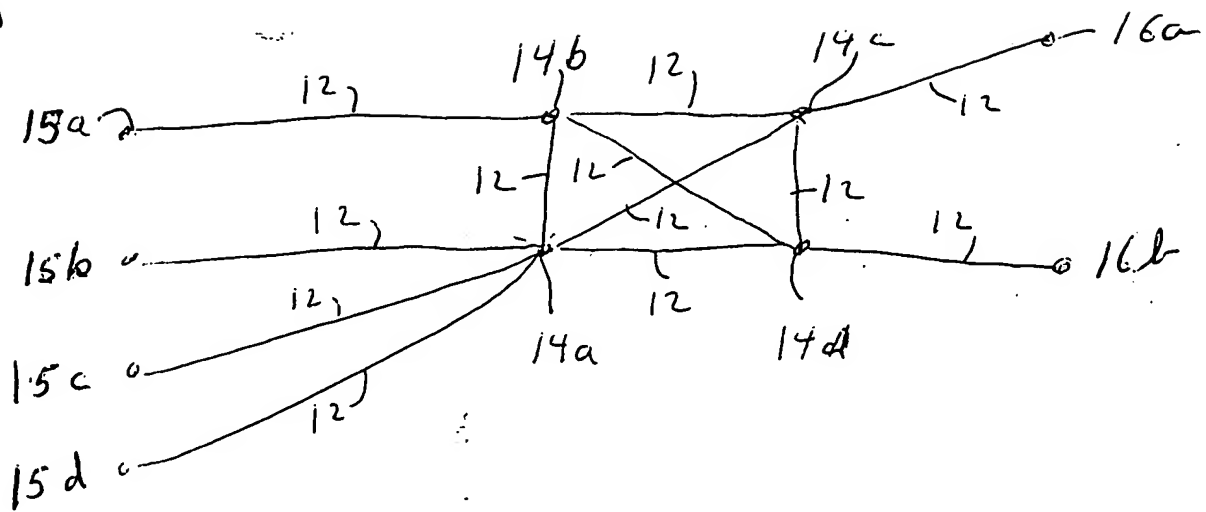


Fig. 1

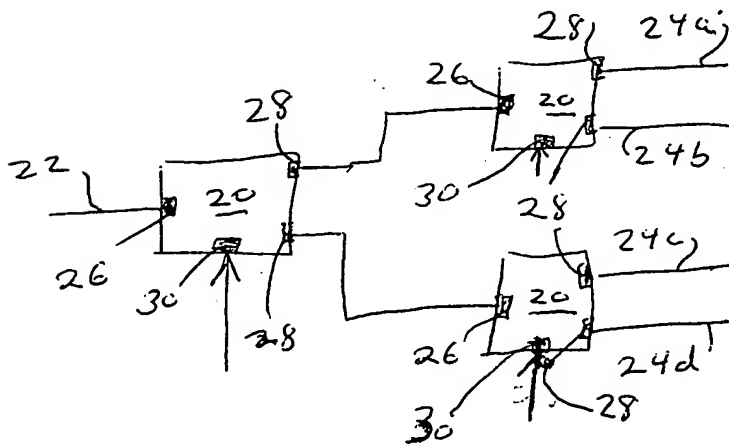


Fig. 2

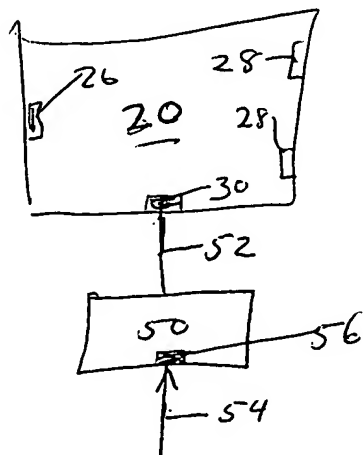


Fig. 3

10035440.022302

10085440 "02222222"

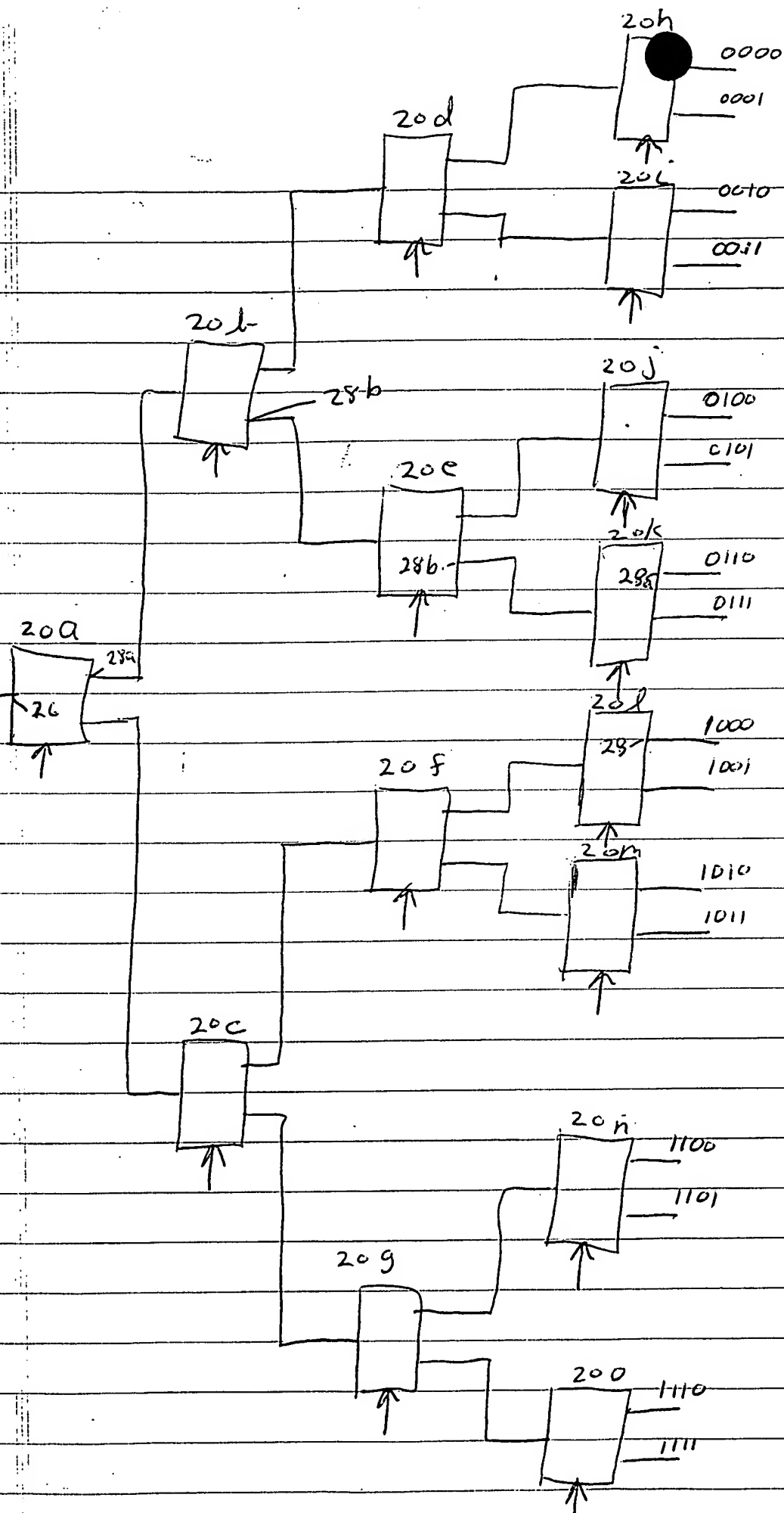


Figure 4

[illegible]

- "1" - top port
- "0" - bottom port
- "x" - "don't care"

Figure 5

1005443-02302

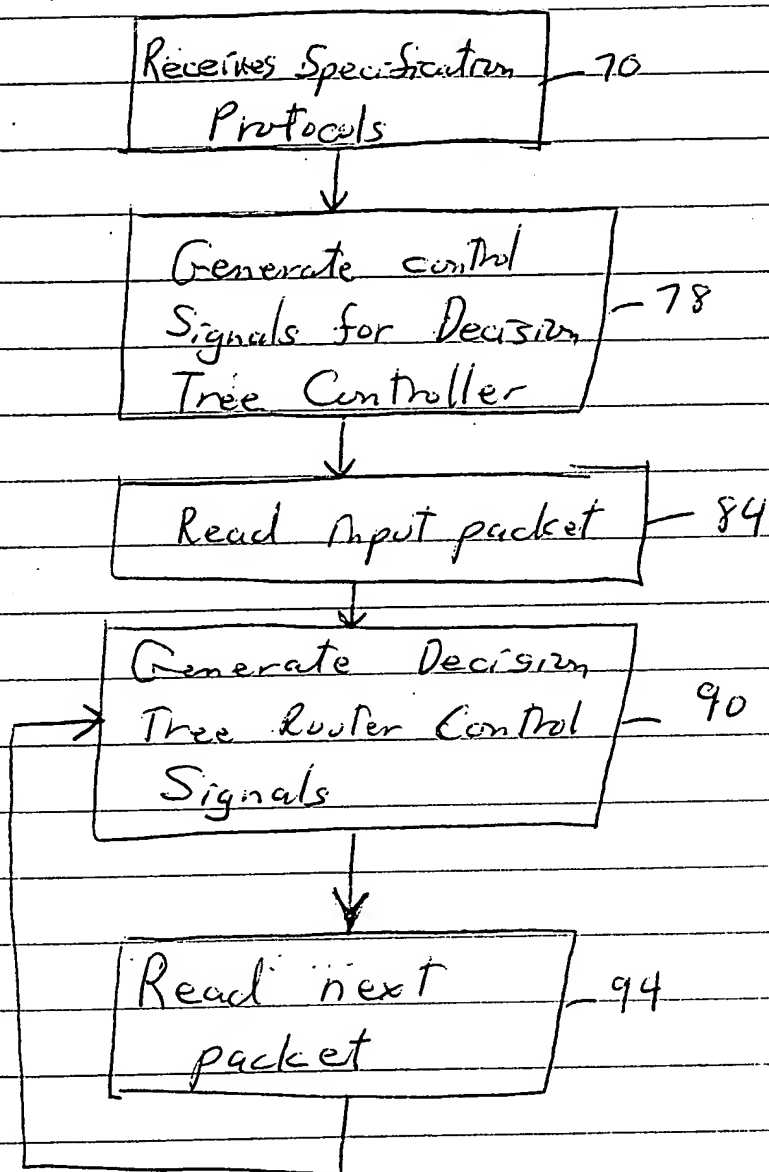


Fig 6

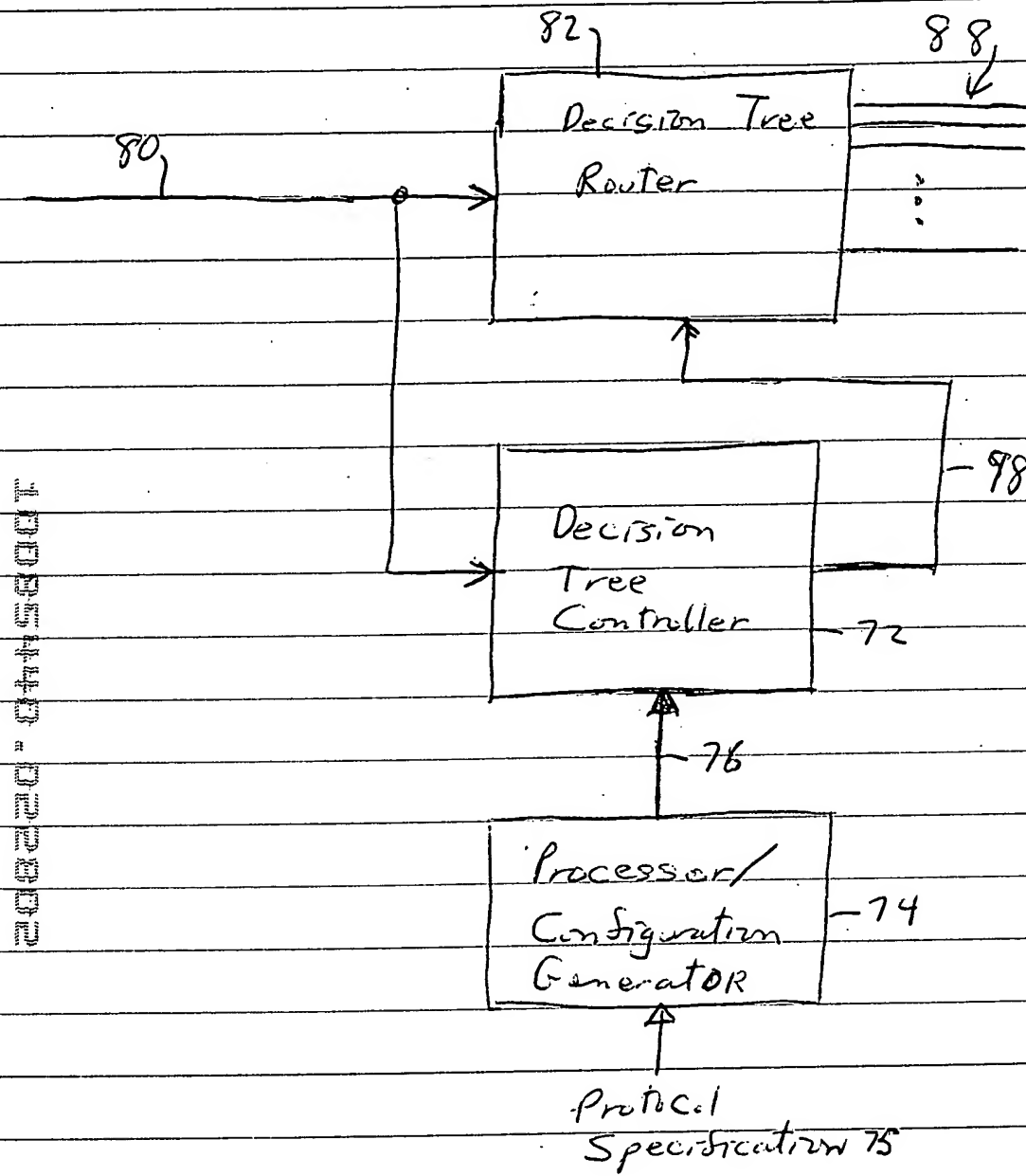


Fig-7